(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 February 2002 (07.02.2002)

PCT

(10) International Publication Number
WO 02/10907 A2

(51) International Patent Classification⁷: G06F 9/00

(21) International Application Number: PCT/US01/15007

(22) International Filing Date: 10 May 2001 (10.05.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/569,602 10 May 2000 (10.05.2000) US

(71) Applicant: **CONVERA CORPORATION** [US/US]; 1921 Gallows Road, Suite 200, Vienna, VA 22182 (US).

(72) Inventors: **RICHARDS, Stefan, N.**; 101 Laurie Meadows Drive, #293, San Mateo, CA 94403 (US). **MILLER, Ron, W.**; 11230 S.E. Highland Loop, Clackamas, OR 97015 (US). **ROZAS, Carlos, V.**; 1534 N.W. Morgan Lane, Portland, OR 97229 (US).

(74) Agents: **SABETT, Randy, V.** et al.; Cooley Godward LLP, Attn.: Patent Group, One Freedom Square, 11951 Freedom Drive, Reston, VA 20191-5601 (US).

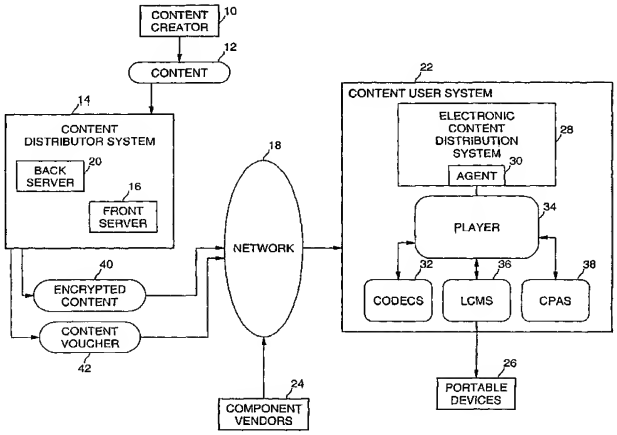
(81) Designated States *(national)*: DE, GB, JP.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD OF REVOKING SOFTWARE COMPONENTS IN A LOOSELY COUPLED CONTENT PROTECTION SYSTEM



(57) **Abstract:** Selective revocation of software components using a key hierarchy in a content protection system. A content distributor creates a content voucher root key, a content voucher signing key, one or more component root keys, and a content voucher. A component vendor creates a vendor root key, a component class key, a component version key, and an object voucher. The content voucher, object voucher and associated software component for processing content may be communicated to a content user system. The keys are used to sign each other in a novel hierarchical arrangement to provide for determination of integrity and authenticity of software components distributed by the component vendor for use on the content user system. The components may be implicitly authorized by the content distributor for use with selected content as a result of the relationships between the keys in the key hierarchy. Revocation of components may be implemented by inserting a revocation list into the content voucher. The revocation list may be checked prior to allowing access to content. Selective revocation of component versions, classes of components, and component vendors may be supported.

Method of Revoking Software Components in a Loosely Coupled Content Protection System

BACKGROUND

1. FIELD

The present invention relates generally to content protection in computer and consumer electronics systems and, more specifically, to revoking authorizations for software components used to render digital content.

2. DESCRIPTION

Physical objects such as compact disks (CDs) or cassette tapes holding entertainment content provide some measure of their own security simply by virtue of the fact that playback of the content is tied to having the physical object present. If one makes a copy (such as by recording music from a CD onto a cassette tape), the quality of the content is degraded. If one wants the highest quality content, one must buy or otherwise obtain an original product. The content distribution industry made it convenient for customers or users to have access to content by making it widely available at high-traffic consumer locations such as record and video stores, malls, and major discount stores. Presently, business to consumer electronic commerce, especially in the area of entertainment content, is growing rapidly on the World Wide Web (WWW) of the Internet. The proliferation of connected personal computers (PCs) and other Internet access devices, the growing bandwidth of the Internet, and better compression techniques are making it possible for content owners to take advantage of the Web as a place to offer digital content for sale and distribution. Many businesses are also increasingly using the Internet as a means to distribute confidential documents, images, video presentations, training and other digital content to employees at geographically dispersed locations.

Thus, the distribution of digital content over the Internet is increasing. With the increasing use of the Internet to buy, sell, or send music, video, documents, images, and other copyrighted or confidential content in digital form comes the need to protect that content from unauthorized use once it is outside the control of the publisher or distributor. Once a user has acquired digital content, the publisher needs to ensure that the usage rights of the content are respected. Since digital rights management (DRM) in computerized systems is typically enforced through software mechanisms, it is important that the software's integrity be checked before content is played back. Also, it is important to ensure that the playback mechanism is authorized by the content owner and that the security software itself has not been tampered with.

One existing technology applicable to digital content protection includes a public key infrastructure (PKI). A PKI enables users of an unsecure public network such as the Internet to securely and privately exchange data through the use of a public and private cryptographic key pair that is obtained and shared through a trusted authority. A PKI provides for digital certificates that can identify individuals or organizations and directory services that can store, and when necessary, revoke them. A PKI assumes the use of public key cryptography for authenticating a message sender or encrypting and decrypting a message. A PKI typically consists of a certificate authority (CA), a registration authority (RA), one or more directories where the certificates (with their public keys) are held, and a certificate management system. The CA is a trusted entity that issues and verifies digital certificates. A digital certificate is an electronic document or file that establishes an entity's credentials when doing business or other transactions on the Web. The certificate might include the public key or information about the public key, as well as other information, such as a name, serial number, expiration dates, and a digital signature of the certificate authority, so that a recipient can verify that the certificate is authentic. Digital certificates may be kept in registries so that authenticated users can look up other users' public keys. A RA is an entity that acts as the verifier for the certificate authority before a digital certificate is issued to a requester. One use of a digital certificate is as an authorization certificate. A CA may delegate some form of authority to a key being signed. For example, a bank could issue an authorization certificate to a customer indicating that a particular key may be used to authorize withdrawal of funds from a specific bank account.

Another technology applicable to content protection is code object signing. Various existing systems have implemented digital signatures for code objects, such as Authenticode, and Java Applet Certificates. Object signing uses techniques of public key cryptography to let users get reliable information about code the users may download from the Internet. When a code object is signed with a valid digital signature, a system may identify the signer and detect tampering with the code object. One approach to detection of tampering is described in a pending patent application entitled "Method and Apparatus for Integrity Verification, Authentication, and Secure Linkage of Software Modules," Serial No. 09/109,472, assigned to the assignee of the present invention.

Previous attempts to safeguard Internet-distributed content at the user site using a PKI and/or code object signing have met several challenges. Solutions have been too costly to deploy and maintain, digital content had to be protected even while it was rendered, and most security solutions eventually need to be renewed. While some protection schemes exist for safeguarding digital content at the content creation and distribution phases, schemes for protecting content at the user site have been problematic. What is needed is a new software security system that acts on behalf of the content owner to ensure the integrity of software, to assert copy control mechanisms, and to ensure that only legitimate users have access to content, all without compromising an enjoyable user experience. Thus, new approaches to protecting content on the client or user system are needed.

BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

Figure 1 is a diagram of an architecture of a key management and content protection system according to an embodiment of the present invention;

Figure 2 is a diagram of a key hierarchy according to an embodiment of the present invention;

Figure 3 is a diagram of an alternate view of a key management and content protection system according to an embodiment of the present invention;

Figure 4 is a flow diagram illustrating content distributor processing according to an embodiment of the present invention;

Figure 5 is a flow diagram illustrating component vendor processing according to an embodiment of the present invention;

Figure 6 is a flow diagram illustrating verification processing according to an embodiment of the present invention;

Figure 7 is a diagram of a key hierarchy exhibiting revocation of a specific version according to an embodiment of the present invention;

Figure 8 is a diagram of a key hierarchy exhibiting revocation of a specific version according to an embodiment of the present invention;

Figure 9 is a diagram of a key hierarchy exhibiting revocation of a specific version according to an embodiment of the present invention; and

Figure 10 is a diagram illustrating an exemplary system used by a content distributor, component vendor, or a content user according to an embodiment of the present invention.

DETAILED DESCRIPTION

An embodiment of the present invention is a method of revoking the authorization of software components using a key hierarchy in a loosely coupled content protection system. The software components may be used to render digital content on a content user system. Revocation of authorization means that the software components can no longer render the content for the user.

Creating a robust end-to-end content distribution system requires the integrity and authenticity of trusted software components running on a user system to be determined before access to content is granted. Public key cryptography may be used as a basis for a key management system supporting content protection according to an embodiment of the present invention. When implementing a key management system as in the present invention for such a content protection system, several issues may be addressed. The security of a cryptographic system usually depends on maintaining the confidentiality of cryptographic keys. The present key management system is designed so as to minimize the risk of compromise of these sensitive keys. For example, the present system does not require one party to disclose sensitive key material to another party in order to express trust relationships between them. The key management system expresses relationships between multiple instances of each participant. For example, a content distributor may want to authorize software components that access the content from several different component vendors. Similarly, a component vendor may have relationships with several different content distributors.

The key management system does not require large amounts of data to be transferred to convey authorization information. It is often useful to allow trust decisions to be transferred to other parties. For example, a content distributor may not want to certify a binary code image for each individual component. Instead, with the present invention a content distributor may delegate authorizations to a component vendor for all components developed by that vendor. Frequently, the trust relationships in the system are not known a priori. The key management system of the present invention handles additions to the trust relationships over time. For example, if a selected digital content title is authorized to be rendered by a particular component (such as a first version of a secure codec, for example), the system does not fail if the user installs a different version of the codec (such as a second, newer version of the same secure codec). Finally, the key management system of the present invention expresses trust relationships that are no longer valid. These revocations may be expressed in a way that does not affect non-compromised components. For example, a content distributor may want to revoke trust of a particular version of a component, without revoking authorization of all components supplied by the content vendor who supplied the particular version.

Reference in the specification to “one embodiment” or “an embodiment” of the present invention means that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, the appearances of the phrase “in one embodiment” appearing in various places throughout the specification are not necessarily all referring to the same embodiment.

Figure 1 is a diagram of an architecture of a key management and content protection system according to an embodiment of the present invention. A content creator 10 is an entity that authors the content. The content creator generates the digital content data 12, which may be stored in a file within a storage medium on a computer system or other digital processing system. For example, a content creator may record and mix music in a recording studio and store the music in a digital form on a storage medium. Generally, the content may be any multimedia content in a digital form, such as audio, video, images, text, music, movies, books, or other data. The content may be sent as either streaming or downloaded content. The format of the content may vary widely depending on the type of content.

It is assumed that the content is created in a trusted environment. A content distributor system 14 is an entity that sells or distributes the content over a communications medium such as the Internet or other computer network. The content distributor system may be controlled by the content owner or may be an authorized independent distributor or reseller of the content. Where the content is digitized music, the content distributor system may represent a music or record company that owns rights to the music or an authorized distributor, such as an on-line retailer. The content distributor system may use one or more server systems to distribute the content to one or more users on demand. In one embodiment, the content distributor system may employ a front server 16 that is coupled to a network 18 such as the Internet to handle queries from a client to obtain Web pages or content. In one embodiment, the protocol supported by the front server is Hyper Text Transport Protocol (HTTP), although other protocols may also be used. The content distributor system may also employ a back server 20, which is not coupled to an outside computer network such as the Internet. Because the back server is "stand-alone" and unconnected to the outside world, it may be considered to be more secure than the front server. In one embodiment, the back server may be a key management workstation used to create and store cryptographic keys employed in the key management system as described further herein.

A content user system 22 is an entity that obtains and consumes the content distributed by content distributor system 14. The content user represents any party seeking to process digital content provided by content distributor system 14, including individual end users, businesses, and other organizations. Content user system 22 operates as a client in a client/server operating model in conjunction with the front server of content distributor system 14. The content user system sends requests for resources (e.g., Web pages, content) to the content distributor system over a network 18 (such as the Internet) and receives data in response to the requests. Network 18 may be any network or series of interconnected networks capable of transporting digital content from the content distributor system to one or more content user systems. For example, network 18 may be a local area network (LAN), a wide area network (WAN), the Internet, or a terrestrial broadcast network such as a satellite communications network. In one embodiment, the content user system comprises at least one of a personal computer (PC) system, an Internet appliance, a set-top box, a handheld computer, a personal digital assistant (PDA), or other computing device. In other embodiments, the content user system may be coupled to the content distributor system through physical distribution of media, such as a floppy disk, a CD-ROM or a digital versatile disk (DVD). One or more component vendors 24 may be coupled to the network for distributing software components for use by a content user in rendering the content or otherwise interacting with the content. Component vendors may be entities independent of the content distributor and content creator who develop software for use in the content user system. In other embodiments, the component vendor may distribute components to the content user system using physical media.

The key management and content protection system of embodiments of the present invention secure content persistently from the time a content user requests a document, book, video, or music to the point it is rendered on the content user system or loaded onto dedicated media players such as portable devices 26. Portable devices may include portable music players (e.g., MP3 players), electronic book readers, or other portable content players. The content protection system uses a protected software module that hides critical code and keys from observation, detects tampering and provides a means of ongoing protection. In one embodiment, the protected software module comprises tamper resistant software. Content is decrypted only after authorizing its use, establishing that the user environment of the content user system 22 is trustworthy while the content is being rendered. The system allows content distributors to renew the security mechanisms, which makes it possible for content distributors to determine the appropriate level of protection by managing the frequency of renewal of trust. Content distributors may distribute many individually encoded security software components. Each content user system may receive an individually encoded component. Content distributors may renew the security software components over the network, helping to reduce the possibility of the components being compromised due to the frequent renewal. The effort needed to compromise the security mechanism is localized to the particular content being protected, so any "hacking" effort has to be re-applied every time new content is delivered and consumed.

The content protection system includes an electronic content distribution (ECD) system 28 resident on content user system 22 to control secure reception and playback of content received from content distributor system 14. The ECD system evaluates content requirements for authorized playback and content user system characteristics to determine if the content user system and its components may be trusted. ECD system 28 includes one or more unique, trusted software components that represent the interests of the content creator or content distributor. This software component, called an agent, acts a custodian of the content creator's interests. In one embodiment, the agent 30 establishes itself as trusted via known tamper-resistant technologies and continuous integrity checking. It then extends the perimeter of trust by continuously checking the integrity of other software components such as plug-ins (e.g., codecs 32) and player 34. Because of this, attempts to tamper with the plug-ins, player or the agent may be detected by the agent 30, and further playback of content by player 34 may be halted. In another embodiment, the agent may operate in an isolated execution mode that provides a measure of security to deter tampering with the agent. In other embodiments, other methods for protecting the agent from tampering may be employed. The agent validates and enforces the conditions that must be fulfilled before the content can be consumed or rendered. These conditions can be anything that the content distributor chooses, consistent with the goal of balancing the protection of the content distributor's rights and the desire to give the content user a satisfying experience in obtaining and rendering the content. For example, the agent might check the content user system 22 for an electronic copy of a purchase receipt or verify a player identifier (ID) or user password. There may be many conditions an agent might verify on the content user system before allowing decryption and rendering of content to occur. After the agent has determined that the legitimate conditions for content usage have been met, the agent decrypts the content so it can be rendered by the player. To reduce the incentive for hacking attempts and minimize the amount of content that is protected by the same security mechanism, the agent may be repeatedly renewed.

Thus, agent 30 interacts with player 34, acting as a controller to determine if and when the content may be rendered. Player 34 may be any software component for processing of digital content. Examples of player 34 for rendering music include RealJukebox™ available from RealNetworks, Inc. and WinAmp™ available from Nullsoft, Inc. Player 34 employs various other software components and plug-ins in order to process the content. Codecs 32 may be used to decode and decompress the decrypted content. One or more licensed compliant modules (LCMs) 36 may be included in the system to interface between the player and portable devices 26. Each LCM may be provided by the manufacturer of the portable device to allow content to flow securely to the portable device from the player. One or more content protection agent service (CPAS) modules 38 may also be included to create an electronic "fingerprint" of the content user system. Any one or more of the components (player, codecs, LCMs, CPAS, and other modules) may be provided by the same or a different component vendor.

The terms voucher, content voucher, object voucher, key voucher, and voucher package used herein may be defined as follows. A voucher may be a digitally signed document. By virtue of it being signed, a voucher's origin can be authenticated, its integrity verified, and the voucher may be non-reputable. There are several types of vouchers. A content voucher contains the "rules" for accessing a piece of digital content. For example, the voucher can specify the characteristics of a trusted software player such that content can only be played by a selected "trusted" software/hardware player combination. Content vouchers are generally created, signed, and distributed by content distributors, and they can be issued for every item of content if needed. An object voucher cryptographically describes a software module. A real world analogy of an object voucher is a driver license. Like a driver license, an object voucher contains a description of the software module but instead of using metrics like height, weight, etc., it uses a cryptographic hash of the software's runtime image. By using object vouchers, software developers can sign not only their components, but also third party components distributed with their applications. A key voucher is used to introduce cryptographic public/private key pairs into a particular domain. Either implicitly or explicitly, a key voucher also grants the key pair certain rights, such as the right to sign content or object vouchers. This process is known as certification. A voucher package may be a data type that contains a "principal" voucher such as a content voucher, and optionally one or more key vouchers. These additional key vouchers form a "chain of trust" that models real world business relationships, such as that of content provider and content distributor. Thus, a content voucher package contains both the rules for accessing content as well as the certification of those rules. This allows for efficient access for verification in a way that is transparent to a software developer.

When the content user system 22 requests content from the content distributor system 14, encrypted content 40 and content voucher 42 may be sent from the content distributor through the network to the content user. When the content is created and mastered by the content creator or the content distributor, a content voucher may be created which is associated with the content and which specifies usage rules for the content. In effect, the content voucher acts as a license for accessing the content. The content voucher may be used by the agent 30 in determining whether to allow player 34 and other components access to the requested content. The content voucher may be assembled with other information to form a content voucher package to be transmitted to the content user system.

In prior art systems, content protection and rendering were often implemented as a single, monolithic system. That is, a content distributor used one type of player which supported a single content format, often with a single security scheme. Once the monolithic security scheme was hacked or broken, all previously distributed content was at risk. Currently, there are many different players supporting different content formats, and the players interact with many different plug-ins provided by different component vendors. In such a scenario, in order to change a plug-in, one would like to not require the content user to return to the content distributor to get a new license for using a new, authorized plug-in with previously authorized content. Additionally, a scheme that protects distributed content even when a single component is hacked would be valuable. Public key cryptography and a key management system according to an embodiment of the present invention may be used to achieve these goals.

Multiple security keys may be used to authorize various content user system components based on the content voucher 42. However, in order to manage the distribution and protection of the keys, a novel key hierarchy is employed. The new key hierarchy allows the content creator and content distributor to decide for which components access is to be provided. Thus, with this key hierarchy system, a new key is used to authorize a new component. The key hierarchy supports revocation of trust of software on a component by component basis, as well as on a per content title basis. That is, authorization of each component and content title in a loosely coupled content protection system may be controlled individually. This provides a fine grained control of authorization and revocation. This provides significant advantages over prior art monolithic content protection systems.

In embodiments of the present invention, the keys in the key hierarchy to be used for content protection may be created by the content distributor systems and component vendors. Various keys may be used to sign content vouchers and software components as will be discussed in further detail below. Embodiments of the present invention work in several stages: content distributor key setup, component vendor authorization, component vendor key setup, component authorization, content licensing, and verification. Tools used to generate cryptographic key pairs, to sign keys, and to create vouchers (e.g., digitally signed documents) are known. Various implementations of such tools may be used with the key hierarchy of the present invention.

Figure 2 is a diagram of a key hierarchy according to an embodiment of the present invention. The diagram shows an example key hierarchy with three different vendors and multiple classes of components. It should be understood that this example is for illustrative purposes and that a key hierarchy according to an embodiment of the present invention may have any number of components, levels and leaf nodes. The example key hierarchy of Figure 2 may be best understood by reference to the architecture diagram of Figure 3 and the flow diagrams of Figures 4 through 6 and accompanying discussion.

Figure 3 shows the interconnections between content distributor system 14, component vendor 24, and content user system 22. In addition, content distributor system 14 may be coupled to agent manufacturing service 44 to request generation of an agent 30 for installation at a content user system 22. Further reference will be made to Figure 3 below.

Figure 4 is a flow diagram illustrating content distributor processing according to an embodiment of the present invention. Content distributor key setup may be as follows. At block 100, content distributor 14 creates a content voucher (CV) root key pair 60. The content voucher root key pair may be used as the root of trust for the key hierarchy and associated loosely coupled content protection system of embodiments of the present invention. The public key of this key pair may be embedded in a trusted checker program 31 that verifies the authenticity of software components on the content user system 22. The trusted checker program 31 may be included in agent 30 downloaded to the content user system within the ECD system 28. The private key of this key pair is of high value. It should be stored on a back server 20, and not on a front server 16 that is accessible to a public network such as the Internet, and thus at risk of unauthorized disclosure. At block 102, the content distributor creates a content voucher (CV) signing key pair 62. This key pair may be used on front server 16 of content distributor 14 to create content vouchers 42 at the time of content purchase. At block 104, the content distributor signs 61 the content voucher signing public key 62 with the content voucher root private key 60. By signing the content voucher signing public key with the content voucher root private key, agent 30 on the content user system (which has the corresponding content voucher root public key) may verify that objects signed by the content voucher signing private key are actually from the content distributor. This exhibits the appropriate delegation of authority. If the content voucher root private key is kept secure, it may also be used to prepare revocation lists at a later point in time.

At block 106, the content distributor creates additional “component-root” key pairs. In another embodiment, the component root key pairs may be created by a trusted delegate of the content distributor. A component root key pair 46 may be referred to by content voucher 42. A component root key may be used to protect a particular component 48 provided to content user system 22 by a component vendor 24. In the example key hierarchy shown in Figure 2, there are two types of components that must be checked for authenticity and integrity (e.g., a codec and a content protection agent service (CPAS) module), so two component root key pairs may be created, a codec root key pair 64 and a CPAS root key pair 66. The private keys of these key pairs should be kept secure on the back server 20. A component root key 46 may be used as a root key for a particular class of components used on content user systems. Other keys based on this root key may comprise a key sub-hierarchy of the main key hierarchy described above. The component may be one of a codec, plug-in, or other software module providing desired functionality of use to the player. Note that the content distributor has created the content voucher root key pair 60, the content voucher signing key pair 62, and multiple component root key pairs 64, 66. Once the content distributor has created the appropriate component root keys, third party component vendors may be authorized.

Vendor authorization proceeds as follows. At block 108, a component vendor 24 creates it’s own component vendor root key pair 50. The component vendor root key may be used as a root key for a vendor key hierarchy 52 which is a subset of the overall key hierarchy for the system. For example, in Figure 2, component vendor 1 creates component vendor 1 root key pair 68, component vendor 2 creates component vendor 2 root key pair 70, component vendor 3 creates component vendor 3 root key pair 72, and so on. Each component vendor has its own vendor key hierarchy for a particular component based on a selected component root key. These keys may be communicated to content distributor system 14 by arrangement with the component vendors. When a content distributor desires to authorize a particular component vendor as a trusted implementer of a particular component, at block 110 the content distributor signs a component vendor’s root public key received from a component vendor with a component root private key corresponding to the desired component. For example, in Figure 2, component vendor 1 root public key 68 may be signed 65 by codec root private key 64, component vendor 2 root public key 70 may be signed 71 by codec root private key 64, and component vendor root public key 72 may be signed 73 by CPAS root private key 66, when content distributor desires to authorize component vendors 1, 2 and 3 as trusted for codecs and CPAS modules, respectively. The signed component vendor's root public key may be communicated back to the component vendor for future use by the component vendor in creating the vendor key hierarchy and object vouchers as discussed further below.

Turning now to Figure 5, a flow diagram illustrating component vendor processing according to an embodiment of the present invention is shown. Each component vendor may set up it's own key hierarchy 52 to distinguish various classes of components that the vendor provides. This supports selective revocation at a later point in time. There may be an arbitrary number of levels of classes and versions of components. Individual vendors may not provide all necessary components to a given functionality, but may aggregate components from other sources. Component vendor key setup proceeds as follows. On a server system at a component vendor 24, a component vendor at block 112 creates a component class key pair for each category or class of component being manufactured or distributed by the vendor. In the example of Figure 2, component vendor 1 creates a codec class key pair 78. At block 114, the component vendor signs the component class keys with the component vendor's root private key. In the example of Figure 2, component vendor 1 signs 76 the codec class public key 78 with component vendor 1 root private key 68.

Component authorization proceeds as follows. When a component vendor desires to authorize a new component to work in the ECD system resident on a content user system 22, component vendor 24 at block 116 creates a component version key pair 54. At block 118, the component vendor signs the component version public key with the component class private key.

This completes the formation of a hierarchy of keys that represents a particular component vendor's trusted product line. In the example of Figure 2, component vendor 1 signs 80 codec version 1.0 public key 82 with codec class private key 78, and component vendor 1 signs 86 codec version 2.0 public key 88 with codec class private key 78. In another variation shown in Figure 2, component vendor 3 signs 90 CPAS version 1.0 public key 92 with component vendor 3 root private key 72. This may be done when no class key is needed, for example, when the vendor only distributes a single component.

Note that in the present key hierarchy, component vendors may sign and distribute new components and associated keys independently of the content distributor. That is, the content distributor does not have to take any actions supporting the distribution of components by the component vendor once the component vendor is authorized. At block 120, the component vendor creates an object voucher 56 and an object voucher package for a particular component being manufactured and distributed. The object voucher package contains information required to authenticate the component on the content user system. The information may include the public portions of the component vendor root key, the component class key, and the component version key. At block 122, the component vendor signs the object voucher for a component with a corresponding component version private key and a cryptographic hash of the component's runtime image. The cryptographic hash may be contained within the object voucher. In the example of Figure 2, component vendor 1 signs 94 codec voucher 96 with codec version 2.0 private key 88, and component vendor 3 signs 98 CPAS voucher 99 with CPAS version 1.0 private key 92. As part of the signing process, the component vendor adds the chain of signed public keys from the hierarchy as additional vouchers to the object voucher packages. This enables content to be accessed by this particular component when verified. Thus, the object voucher package contains the content distributor component root public key, and the component version public key of the key hierarchy corresponding to the component object. The object voucher also contains the cryptographic signature of each public key, signed by its parent's private key (e.g., the component version public key, signed by the component class private key). The object voucher 56 may be signed by the component vendor using an object signing tool 58. At some point in time prior to content distribution, the content user system obtains relevant software components such as player 34, codecs 32, LCMs 36, CPAS modules 38 from component vendors. In one embodiment, the components may be obtained over a network such as the Internet from each component vendor. When such software is installed, associated object vouchers 56 for each component created by the component vendors may also be included in the installation on the content user system.

Content licensing proceeds as follows. At block 124, at the time a selected content title is vended or otherwise distributed by content distributor system 14 to content user system 22, the content distributor creates a content voucher package containing the content voucher 42 to specify all usage rules for that content title and one or more key vouchers. The usage rules may specify the components authorized to process the content. The key voucher may include the content voucher signing public key signed by the content voucher root private key. The component root public key 46 created at block 106 may be inserted into the content voucher 42. The content voucher may be signed 63 by the content voucher signing private key 62. The content voucher package may then be communicated to the content user system.

Verification processing proceeds as follows. Figure 6 is a flow diagram illustrating verification processing on the content user system according to an embodiment of the present invention. At the time of access to the content, a trusted checker module 31 within agent 30 on the content user system determines the trust of the other components 48 in the system based on content voucher 42 within the content voucher package and the object voucher 56. At block 126, the checker 31 checks the integrity of the signed content voucher 42 using the content voucher signing public key 62 contained in the content voucher. At block 128, the checker module determines the trust of the content voucher signing public key 62 by checking the signature of the content voucher signing public key using the content voucher root public key 60 embedded in the agent 30. At block 130, the checker parses the component root keys 46 included in the content voucher 42. At block 132, the checker determines the integrity of a selected object voucher 56 corresponding to a component 48 using a component version public key 54 contained in the object voucher.

At block 134, the trust of the component version public key 54 may be determined by checking the signatures of a chain of keys up to one of the component keys parsed from the content voucher at block 130. At block 136, the integrity of component 48 (e.g., a player, a codec, a plug-in, etc.) may be checked using information parsed from the corresponding object voucher 56. At block 138, when the integrity and authenticity of the component have been established using the key hierarchy of the present invention, the content user system may be allowed to decrypt and render the content using appropriate content protection keys contained in the content voucher.

The configuration of the key hierarchy of the present invention provides at least several benefits for protecting content and components. The fact that the key hierarchy is based on public key cryptography allows authorizations to be completed without private key material being exposed. In addition, the use of the content voucher signing key by the content distributor ensures that a high value root key is not needed for an on-line transaction. This allows the root key to be stored and managed in a more secure environment. Since authorizations are expressed by a key hierarchy, flexibility is built into the system. Each content distributor can authorize multiple component vendors under this hierarchy. Similarly, a component vendor can support authorization of multiple classes of components and multiple versions of components in the vendor's portfolio. With respect to efficient data representations, as opposed to having an itemized list of authorized components, the use of the present key hierarchy allows authorization of potentially thousands of components to be expressed in very small data structures.

Once the content distributor signs a component vendor root key, the component vendor can build and authorize components on behalf of the content distributor. This allows content distributors to remain out of the development life cycle of component vendors. Once the content distributor signs the component vendor root key, the component vendor can continue to build new versions of components that will work with the existing content vouchers. This is in contrast to a prior art system where authorized components were explicitly listed in content vouchers. In that case, new content vouchers have to be issued to work with new components. Embodiments of the present invention provide a clear advantage in this regard.

Regarding revocation, the fact that a component vendor creates a hierarchy of keys for a product line allows a content distributor to revoke a vendor at the component version, component class, or vendor level. If only a single key was used to sign all components from a vendor, a revocation would affect all components created by that vendor. Because the content voucher signing key is used to sign component class keys at content voucher creation time, the component certification key hierarchies can be disparate from the rights key hierarchy, and only bound together for a particular licensing transaction. This enables the content distributor to easily switch component class keys in the event of a key compromise, while instantaneously invalidating the compromised hierarchy. Replacing the component class key in a new content voucher achieves revocation of a previously valid component class key for new content, without the use of a revocation list.

Revocation of authorization may be needed in any one of several instances. For example, if it is discovered that a malicious user has “broken” or “hacked” the content protection system, the content distributor may want to revoke access to distributed content. If any key distributed to a component vendor or content user system is accidentally disclosed, authorization of components affected by the disclosed key should be revoked. If a component root key is lost or disclosed, authorization of components corresponding to the component root key should be revoked. If a holder of a private key becomes malicious for some reason, authorization of components should be revoked.

As discussed above, the signed content voucher specifies the content rights granted to the user system. The content voucher also specifies trusted components comprising a system in which content may be rendered. Components may be specified based on an inclusive model, meaning that any component issued under a trusted key specified for that class of component (e.g., a component root key) is automatically trusted. With embodiments of the present invention, a component may be revoked by including a revocation list in the content voucher. The revocation list may contain information pertaining to one or more components which are no longer authorized. The revocation list may include at least one of the cryptographic hash of the component's binary image, the component root key associated with the components, or any keys in its key hierarchy, depending on the scope of the revocation desired. By doing this, at the time of determining authorization to access the content, the revocation list may be retrieved from the content voucher and checked against the key hierarchy used to sign a given component. If a key needed to verify authorization of the component is in the revocation list, authorization fails. Thus, embodiments of the present invention bind the revocation action to a single operating context within the content protection system and does not affect any other context.

In embodiments of the present invention, at least three different levels of revocation may be supported. These three levels exhibit the flexibility of the present invention in supporting revocation of components. First, revocation of a specific version of a component may be supported. For example, a content distributor may revoke authorization of codec version 2.0 88 and associated codec voucher 96 as shown in Figure 7. Revocation of the key that was used to sign the codec voucher specifies that the agent executing on the content user system will not trust or use this component version from the component vendor. The agent instead will gracefully fail on its attempt to use the component, thereby preventing access to the content by the revoked component. As the example diagram shows, the key hierarchy for the codec component includes a content distributor's component root key (e.g., codec root key pair 64) that it uses to sign the component vendor's root keys (e.g., component vendor 1 root key pair 68). The component vendor creates hierarchies according to the granularity of its products and versions. The more keys in the hierarchy, the better the granularity of revocation that may be achieved.

Second, revocation of a specific class of components may be supported. For example, a content distributor may revoke the class of codecs as shown in Figure 8. Revocation of the key that was used to sign this particular class of component provided by a specific component vendor specifies that the agent will not trust or use any component signed under this key. The agent instead will gracefully fail on its attempt to use any component of this class from the component vendor to access the content.

Third, revocation of a specific component vendor may be supported. For example, a content distributor may revoke all components supplied by a particular component vendor as shown in Figure 9. Revocation of the key that was used by the component vendor to sign classes of components specifies that the agent will not trust or use any component signed under this key. The agent instead will gracefully fail on its attempt to use any component from this component vendor to access the content.

Content distributor actions supporting revocation according to an embodiment of the present invention are shown in Figure 10. At block 200, a break may be detected in the content protection system. For example, the content distributor may discover or be informed that one or more private keys of the key hierarchy have been disclosed or "hacked". In such instances, the content distributor may communicate with the component vendor to identify specific component classes or versions of components to revoke. If a particular component vendor is considered to be suspect by the content distributor, all components supplied by the component vendor may be revoked. At block 202, the content distributor creates a list of revoked public keys from the key hierarchy due to the break. Since the key hierarchy is a tree, only the highest level key to be revoked need be specified (e.g., the root node of the sub-tree of all keys in the hierarchy to be revoked) instead of all such keys including all leaf nodes of the sub-tree in the hierarchy. Next, at block 204, the content distributor creates a data structure to embed in the content voucher. The data structure contains at least the revoked public keys. When the content distributor creates the content voucher at block 124 of Figure 5 prior to vending the content to the user, the content distributor may now insert the revocation list into the content voucher. Hence, revocation of versions and classes of components, and component vendors, may be implemented, at least in part, by vending content to users. This action protects new content that is being distributed from earlier breaks in the security of the content protection system.

When the checker within the agent is verifying authorization of a selected component as shown in the steps of Figure 6, the checker may now also check the revocation list contained in the content voucher at block 206 of Figure 10. If a particular component vendor public key, component class public key, or component version public key is in the revocation list, the checker causes a failure of the authorization process and the content may not be accessed by the selected component being checked. Otherwise, if the selected key is not on the revocation list, it is presumed that the component has not been revoked by the content distributor and the usual integrity verification and authentication operations relating to the content and object vouchers may be executed.

Thus, the combination of the revocation list with the novel key hierarchy of the present invention allows a content distributor to selectively revoke components, classes of components, and component vendors. This invention provides at least several advantages. In current systems, selective revocation of software components is not supported. Current systems typically only support revocation of entire classes of devices. That is, in known public key infrastructures, revocations are issued in the form of signed lists, independent of any contextual binding. The present invention binds the revocation list directly to the usage context. This means that revocations are applied locally, in the context they were intended, rather than globally. With the present invention, components may be revoked when new content is delivered, without removing the user's existing rights to old content. This leads to a better user experience. It also enables a content distributor, collecting content from many content owners or creators, to enforce the usage specified by each content owner discretely. This feature cannot be enabled by a typical certificate revocation list of the prior art. The present invention also allows individual components to be revoked, without removing a component vendor's ability to manufacture and certify components with their existing certified keys. As components are shown to expose security weaknesses, they can be selectively revoked for all future content without removing the vendor's capability to fix vulnerabilities and release new products that work with new content without the need to get a new key or signature from the content distributor. The system also allows revocation to occur at various granularities, from a single component to a whole class of components. This allows content distributors to be as selective as they want to be in choosing components to work with on a per context basis. Also, because of the organization of this novel key hierarchy and the present invention, component vendors cannot revoke components issued by competing component vendors. The present invention also allows content owners to independently decide which components should be considered to be trustworthy for their content (i.e., they control the hierarchy for their content).

In the preceding description, various aspects of the present invention have been described. For purposes of explanation, specific numbers, systems and configurations were set forth in order to provide a thorough understanding of the present invention. However, it is apparent to one skilled in the art having the benefit of this disclosure that the present invention may be practiced without the specific details. In other instances, well-known features were omitted or simplified in order not to obscure the present invention.

Embodiments of the present invention may be implemented in hardware or software, or a combination of both. However, embodiments of the invention may be implemented as computer programs executing on programmable systems comprising at least one processor, a data storage system (including volatile and non-volatile memory and/or storage elements), at least one input device, and at least one output device. Program code may be applied to input data to perform the functions described herein and generate output information. The output information may be applied to one or more output devices, in known fashion. For purposes of this application, a processing system embodying the playback device components includes any system that has a processor, such as, for example, a digital signal processor (DSP), a microcontroller, an application specific integrated circuit (ASIC), or a microprocessor.

The programs may be implemented in a high level procedural or object oriented programming language to communicate with a processing system. The programs may also be implemented in assembly or machine language, if desired. In fact, the invention is not limited in scope to any particular programming language. In any case, the language may be a compiled or interpreted language.

The programs may be stored on a removable storage media or device (e.g., floppy disk drive, read only memory (ROM), CD-ROM device, flash memory device, digital versatile disk (DVD), or other storage device) readable by a general or special purpose programmable processing system, for configuring and operating the processing system when the storage media or device is read by the processing system to perform the procedures described herein. Embodiments of the invention may also be considered to be implemented as a machine-readable storage medium, configured for use with a processing system, where the storage medium so configured causes the processing system to operate in a specific and predefined manner to perform the functions described herein.

An example of one such type of processing system is shown in Figure 11, however, other systems may also be used and not all components of the system shown are required for the present invention. Sample system 400 may be used, for example, to execute the processing for embodiments of the key hierarchy and content protection system, in accordance with the present invention, such as the embodiment described herein. Sample system 400 is representative of processing systems based on the PENTIUM®II, PENTIUM® III, and CELERON™ microprocessors available from Intel Corporation, although other systems (including personal computers (PCs) having other microprocessors, engineering workstations, other set-top boxes, and the like) and architectures may also be used.

Figure 11 is a block diagram of a system 400 of one embodiment of the present invention. The system 400 includes a processor 402 that processes data signals. Processor 402 may be coupled to a processor bus 404 that transmits data signals between processor 402 and other components in the system 400.

System 400 includes a memory 406. Memory 406 may store instructions and/or data represented by data signals that may be executed by processor 402. The instructions and/or data may comprise code for performing any and/or all of the techniques of the present invention. Memory 406 may also contain additional software and/or data (not shown). A cache memory 408 may reside inside processor 402 that stores data signals stored in memory 406.

A bridge/memory controller 410 may be coupled to the processor bus 404 and memory 406. The bridge/memory controller 410 directs data signals between processor 402, memory 406, and other components in the system 400 and bridges the data signals between processor bus 404, memory 406, and a first input/output (I/O) bus 412. In this embodiment, graphics controller 413 interfaces to a display device (not shown) for displaying images rendered or otherwise processed by the graphics controller 413 to a user.

First I/O bus 412 may comprise a single bus or a combination of multiple buses. First I/O bus 412 provides communication links between components in system 400. A network controller 414 may be coupled to the first I/O bus 412. In some embodiments, a display device controller 416 may be coupled to the first I/O bus 412. The display device controller 416 allows coupling of a display device to system 400 and acts as an interface between a display device (not shown) and the system. The display device receives data signals from processor 402 through display device controller 416 and displays information contained in the data signals to a user of system 400.

A second I/O bus 420 may comprise a single bus or a combination of multiple buses. The second I/O bus 420 provides communication links between components in system 400. A data storage device 422 may be coupled to the second I/O bus 420. A keyboard interface 424 may be coupled to the second I/O bus 420. A user input interface 425 may be coupled to the second I/O bus 420. The user input interface may be coupled to a user input device, such as a remote control, mouse, joystick, or trackball, for example, to provide input data to the computer system. An audio controller 427 may be coupled to the second I/O bus for handling processing of audio signals through one or more loudspeakers (not shown). A bus bridge 428 couples first I/O bridge 412 to second I/O bridge 420.

Embodiments of the present invention are related to the use of the system 400 as a content distributor, component vendor, or content user system. According to one embodiment, such processing may be performed by the system 400 in response to processor 402 executing sequences of instructions in memory 404. Such instructions may be read into memory 404 from another computer-readable medium, such as data storage device 422, or from another source via the network controller 414, for example. Execution of the sequences of instructions causes processor 402 to execute content protection processing according to embodiments of the present invention. In an alternative embodiment, hardware circuitry may be used in place of or in combination with software instructions to implement embodiments of the present invention. Thus, the present invention is not limited to any specific combination of hardware circuitry and software.

The elements of system 400 perform their conventional functions in a manner well known in the art. In particular, data storage device 422 may be used to provide long-term storage for the executable instructions and data structures for embodiments of the content protection system in accordance with the present invention, whereas memory 406 is used to store on a shorter term basis the executable instructions of embodiments of the content protection system in accordance with the present invention during execution by processor 402.

While this invention has been described with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments of the invention, which are apparent to persons skilled in the art to which the inventions pertains are deemed to lie within the spirit and scope of the invention.

CLAIMS

What is claimed is:

1. A method of selectively revoking authorization of software components by a content distributor in a content protection system comprising:

creating a list of at least one key corresponding to at least one component for which authorization is to be revoked, the at least one key belonging to a key hierarchy used in the content protection system, the key hierarchy comprising a tree having a component vendor root key corresponding to a component vendor as a root of the tree; and

inserting the list into a content voucher, the content voucher specifying rules for access to selected content by the at least one component.

2. The method of claim 1, wherein the at least one key is a leaf of the tree, the at least one component being resident on a user system in the content protection system.

3. The method of claim 1, further comprising sending the content voucher to the user system.

4. The method of claim 3, further comprising causing the list in the content voucher to be checked by a software module on the user system to verify authorization of the at least one component and revoking authorization of the at least one component for accessing the content when the at least one key is on the list.

5. The method of claim 4, wherein when the list includes the component vendor root key, revoking authorization of all classes and all components distributed by the component vendor.

6. The method of claim 5, wherein the key hierarchy comprises at least one class key corresponding to a selected class of components, the class key being a root of a sub-tree of the tree, and wherein when the list includes a class key, revoking authorization of all components of the class distributed by the component vendor.

7. The method of claim 4, further comprising revoking authorization of a component when the list includes a corresponding component key.

8. The method of claim 4, further comprising revoking authorization of a component when the list includes a hash of the component.

9. An article comprising: a storage medium having a plurality of machine readable instructions, wherein when the instructions are executed by a processor, the instructions provide for selective revocation of authorization of a software component by a content distributor in a content protection system, the instructions including creating a list of at least one key corresponding to at least one component for which authorization is to be revoked, the at least one key belonging to a key hierarchy used in the content protection system, the key hierarchy comprising a tree having a component vendor root key corresponding to a component vendor as a root of the tree; and inserting the list into a content voucher, the content voucher specifying rules for access to selected content by the at least one component.

10. The article of claim 9, wherein the at least one key is a leaf of the tree, the at least one component being resident on a user system in the content protection system

11. The article of claim 10, further comprising instructions for sending the content voucher to the user system.

12. The article of claim 11, further comprising instructions for causing the list in the content voucher to be checked by a software module on the user system to verify authorization of the at least one component and revoking authorization of the at least one component for accessing the content when the at least one key is on the list.

13. The article of claim 12, wherein when the list includes the component vendor root key, further comprising instructions for revoking authorization of all classes and all components distributed by the component vendor.

14. The article of claim 13, wherein the key hierarchy comprises at least one class key corresponding to a selected class of components, the class key being a root of a sub-tree of the tree, and wherein when the list includes a class key, further comprising instructions for revoking authorization of all components of the class distributed by the component vendor.

15. The article of claim 12, further comprising instructions for revoking authorization of a component when the list includes a corresponding component key.

16. The article of claim 12, further comprising instructions for revoking authorization of a component when the list includes a hash of the component.

17. A content protection system providing selective revocation of software components comprising:

a key hierarchy comprising a tree having a component vendor root key corresponding to a component vendor as a root of the tree and at least one key as a leaf of the tree corresponding to at least one component for which authorization is to be revoked, the at least one component being resident on a user system in the content protection system;

a content voucher specifying rules for access to selected content by the at least one component, the content voucher including a list having the at least one key corresponding to the at least one component for which authorization is to be revoked; and

a content distributor to create the content voucher and send the content voucher to the user system.

18. The system of claim 17, further comprising a software module resident on the user system to check the list in the content voucher to verify authorization of the at least one component and revoke authorization of the at least one component for accessing the content when the at least one key is on the list.

19. The system of claim 18, wherein when the list includes the component vendor root key, the software module revokes authorization of all classes and all components distributed by the component vendor.

20. The system of claim 19, wherein the key hierarchy comprises at least one class key corresponding to a selected class of components, the class key being a root of a sub-tree of the tree, and wherein when the list includes a class key, the software module revokes authorization of all components of the class distributed by the component vendor.

21. The system of claim 20, wherein the software module comprises a tamper-resistant module.

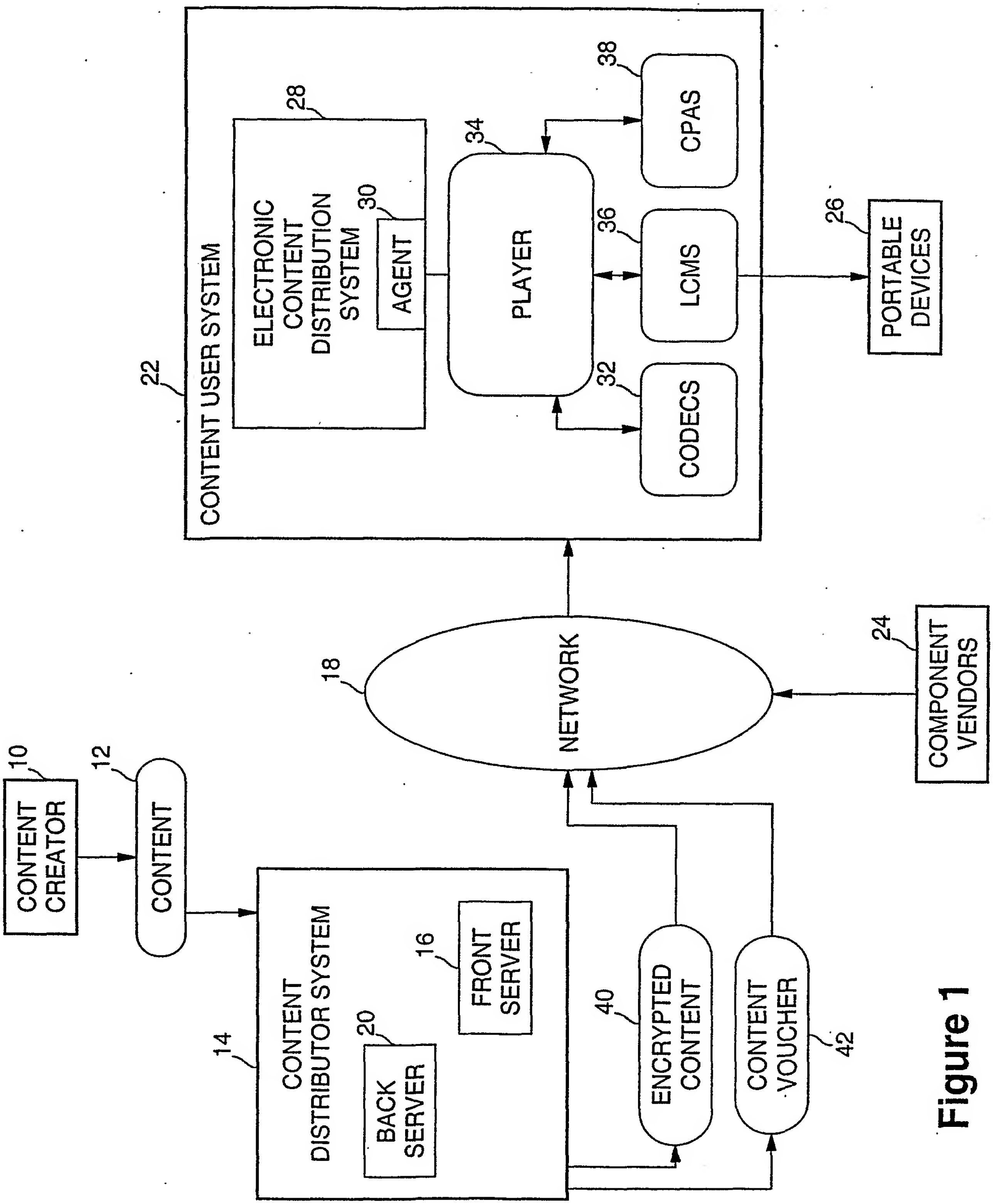

CONTENT CREATOR
10
CONTENT
12
CONTENT DISTRIBUTOR SYSTEM
14
BACK SERVER
20
FRONT SERVER
16
ENCRYPTED CONTENT
40
CONTENT VOUCHER
42
NETWORK
18
COMPONENT VENDORS
24
CONTENT USER SYSTEM
22
ELECTRONIC CONTENT DISTRIBUTION SYSTEM
28
AGENT
30
PLAYER
34
CODECS
32
LCMS
36
CPAS
38
PORTABLE DEVICES
26

Figure 1

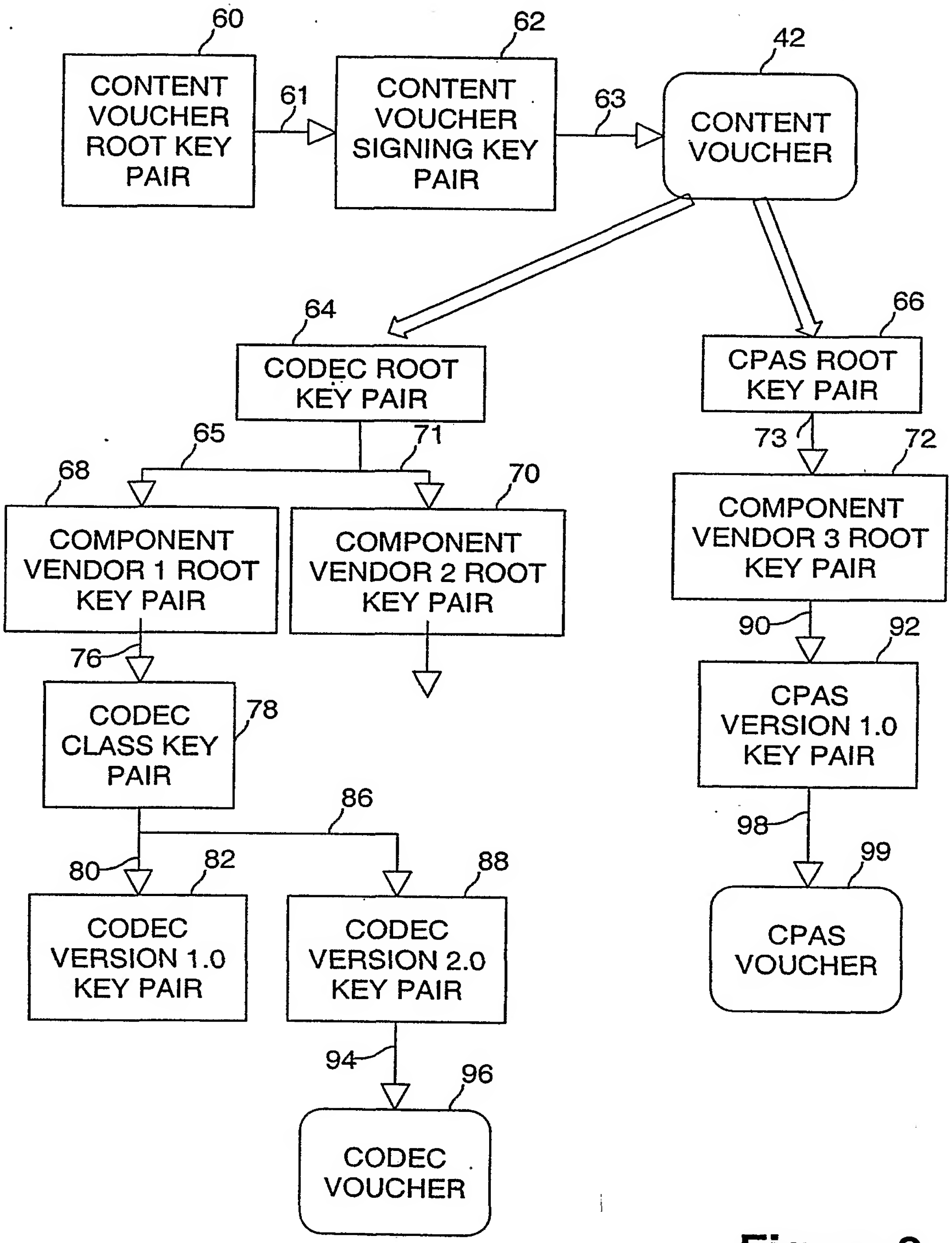

60
CONTENT VOUCHER ROOT KEY PAIR
61
62
CONTENT VOUCHER SIGNING KEY PAIR
63
42
CONTENT VOUCHER
64
CODEC ROOT KEY PAIR
66
CPAS ROOT KEY PAIR
65
71
68
70
73
72
COMPONENT VENDOR 1 ROOT KEY PAIR
COMPONENT VENDOR 2 ROOT KEY PAIR
COMPONENT VENDOR 3 ROOT KEY PAIR
76
90
92
CODEC CLASS KEY PAIR
78
CPAS VERSION 1.0 KEY PAIR
86
98
99
80
82
88
CODEC VERSION 1.0 KEY PAIR
CODEC VERSION 2.0 KEY PAIR
CPAS VOUCHER
94
96
CODEC VOUCHER

Figure 2

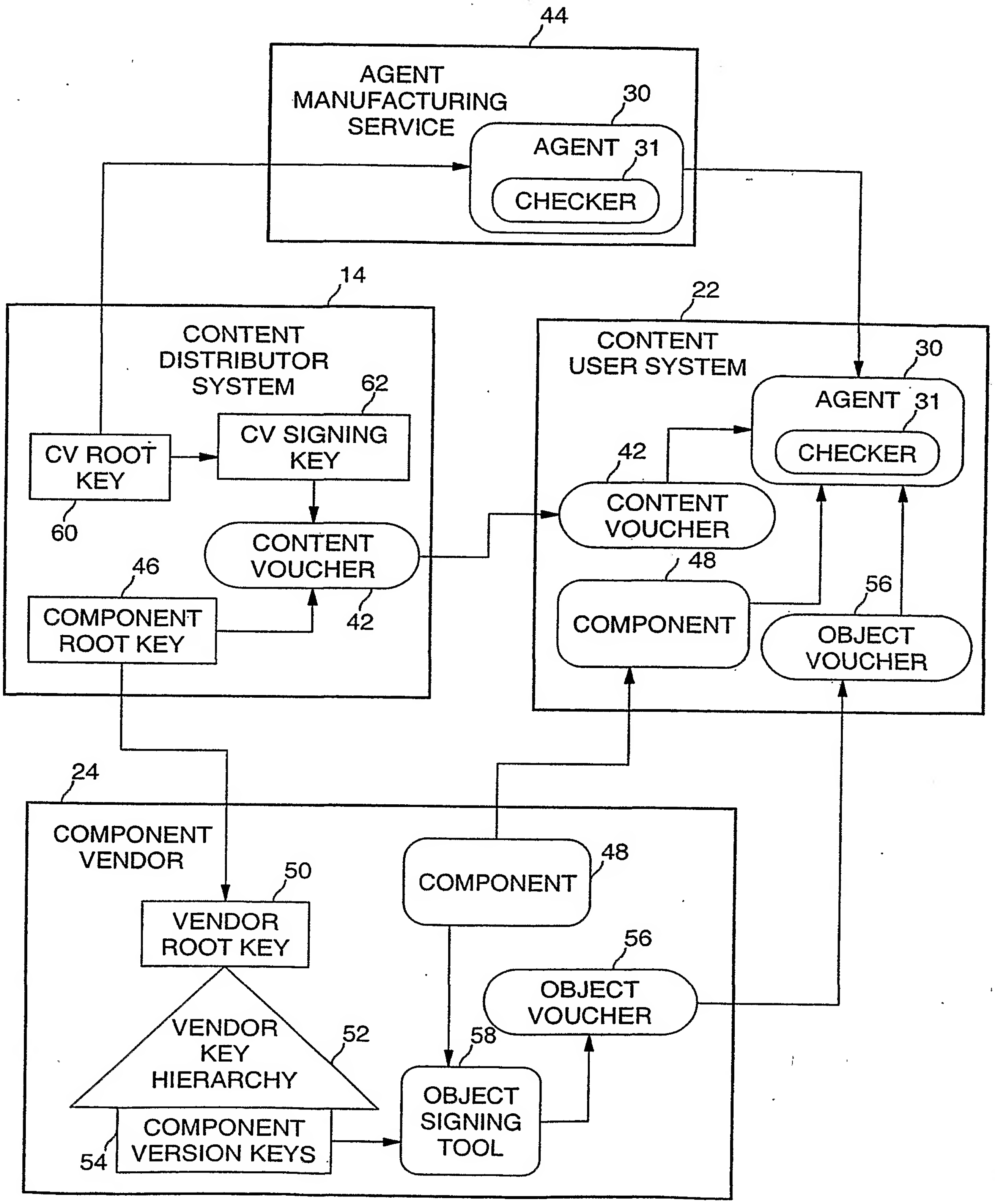

44
AGENT MANUFACTURING SERVICE
30
AGENT 31
CHECKER
14
CONTENT DISTRIBUTOR SYSTEM
62
CV ROOT KEY
CV SIGNING KEY
60
CONTENT VOUCHER
46
COMPONENT ROOT KEY
42
22
CONTENT USER SYSTEM
30
AGENT 31
CHECKER
42
CONTENT VOUCHER
48
COMPONENT
56
OBJECT VOUCHER
24
COMPONENT VENDOR
50
VENDOR ROOT KEY
VENDOR KEY HIERARCHY
52
COMPONENT VERSION KEYS
54
COMPONENT
48
56
OBJECT VOUCHER
58
OBJECT SIGNING TOOL

Figure 3

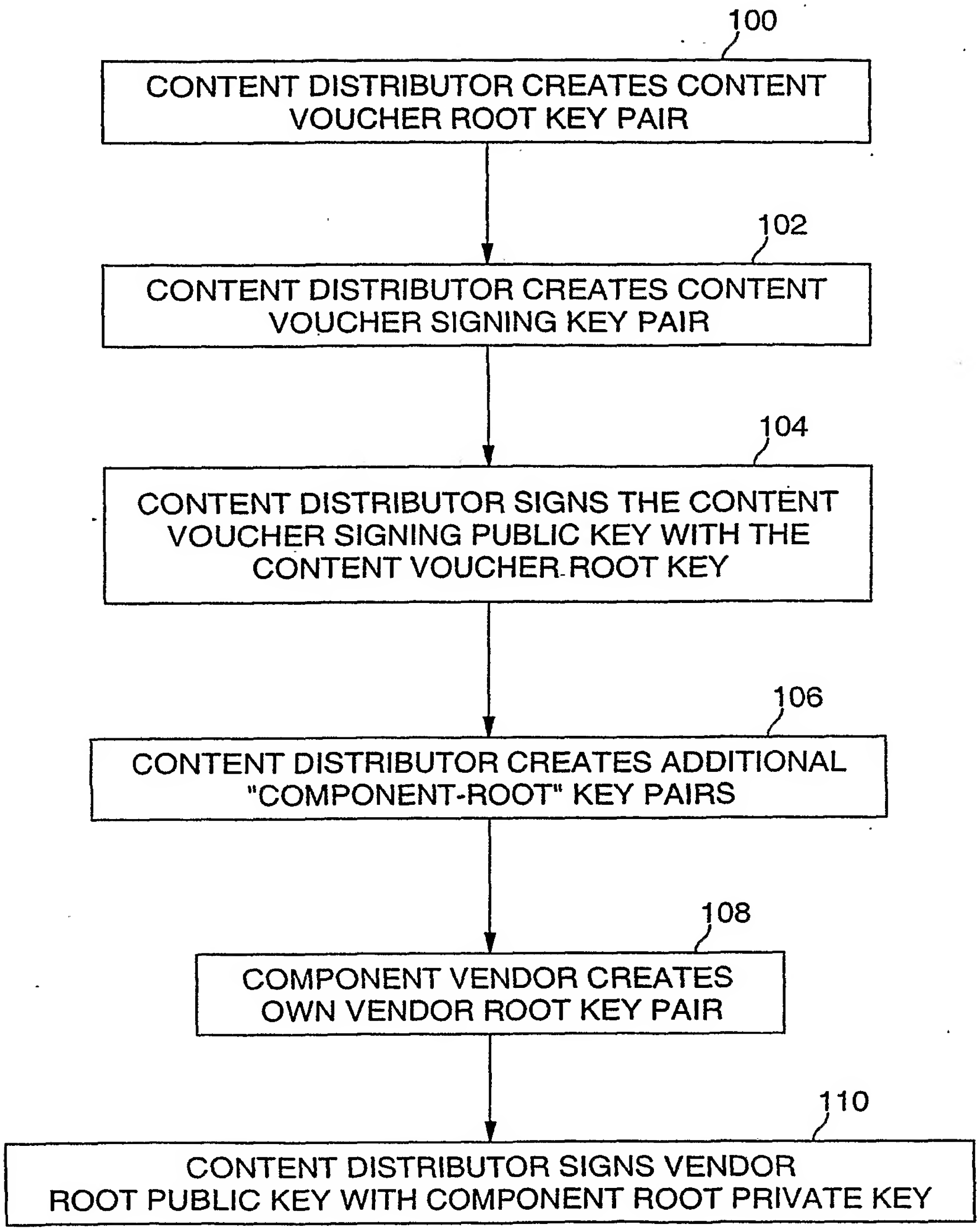

100
CONTENT DISTRIBUTOR CREATES CONTENT VOUCHER ROOT KEY PAIR
102
CONTENT DISTRIBUTOR CREATES CONTENT VOUCHER SIGNING KEY PAIR
104
CONTENT DISTRIBUTOR SIGNS THE CONTENT VOUCHER SIGNING PUBLIC KEY WITH THE CONTENT VOUCHER ROOT KEY
106
CONTENT DISTRIBUTOR CREATES ADDITIONAL "COMPONENT-ROOT" KEY PAIRS
108
COMPONENT VENDOR CREATES OWN VENDOR ROOT KEY PAIR
110
CONTENT DISTRIBUTOR SIGNS VENDOR ROOT PUBLIC KEY WITH COMPONENT ROOT PRIVATE KEY

Figure 4

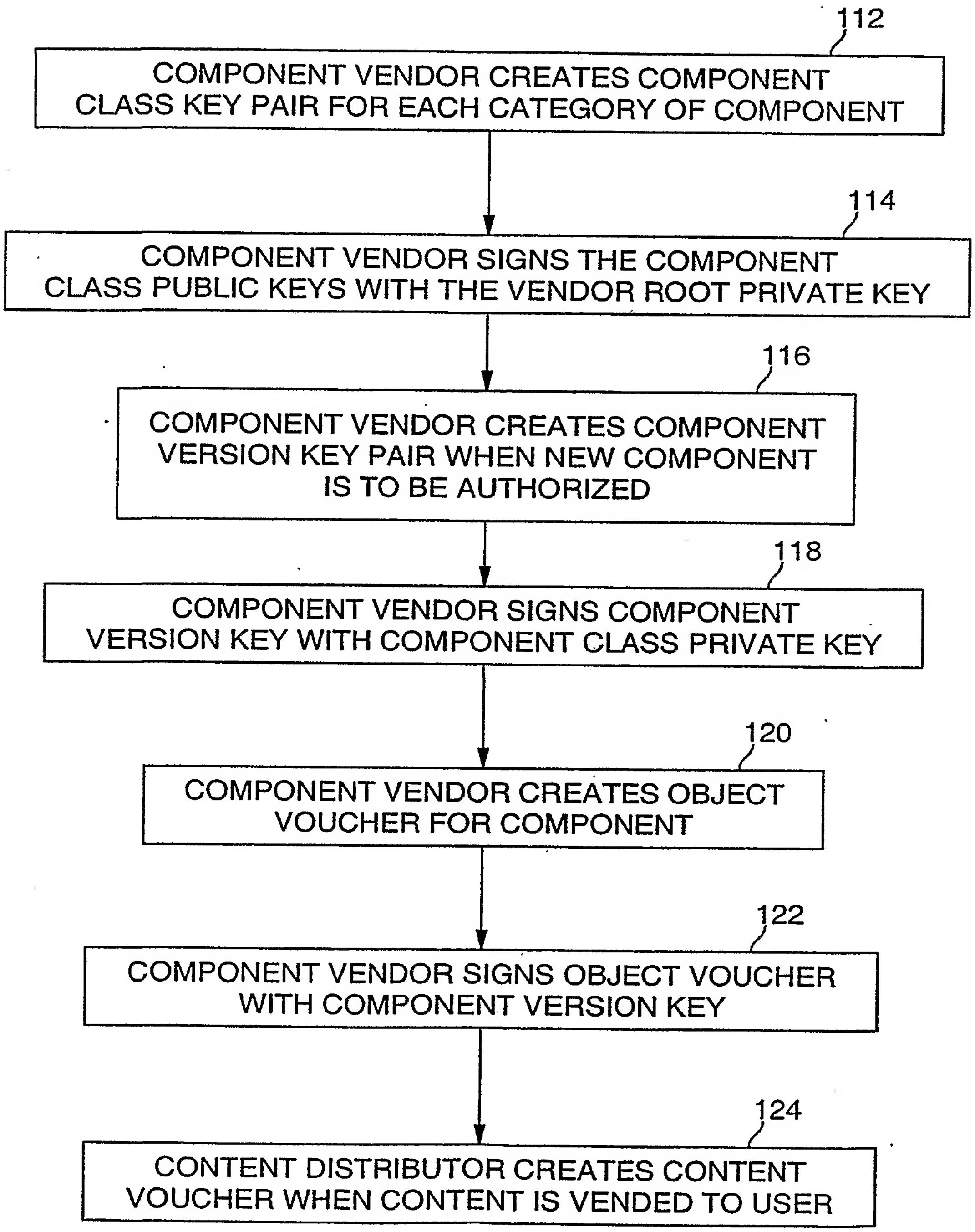

112
COMPONENT VENDOR CREATES COMPONENT CLASS KEY PAIR FOR EACH CATEGORY OF COMPONENT
114
COMPONENT VENDOR SIGNS THE COMPONENT CLASS PUBLIC KEYS WITH THE VENDOR ROOT PRIVATE KEY
116
COMPONENT VENDOR CREATES COMPONENT VERSION KEY PAIR WHEN NEW COMPONENT IS TO BE AUTHORIZED
118
COMPONENT VENDOR SIGNS COMPONENT VERSION KEY WITH COMPONENT CLASS PRIVATE KEY
120
COMPONENT VENDOR CREATES OBJECT VOUCHER FOR COMPONENT
122
COMPONENT VENDOR SIGNS OBJECT VOUCHER WITH COMPONENT VERSION KEY
124
CONTENT DISTRIBUTOR CREATES CONTENT VOUCHER WHEN CONTENT IS VENDED TO USER

Figure 5

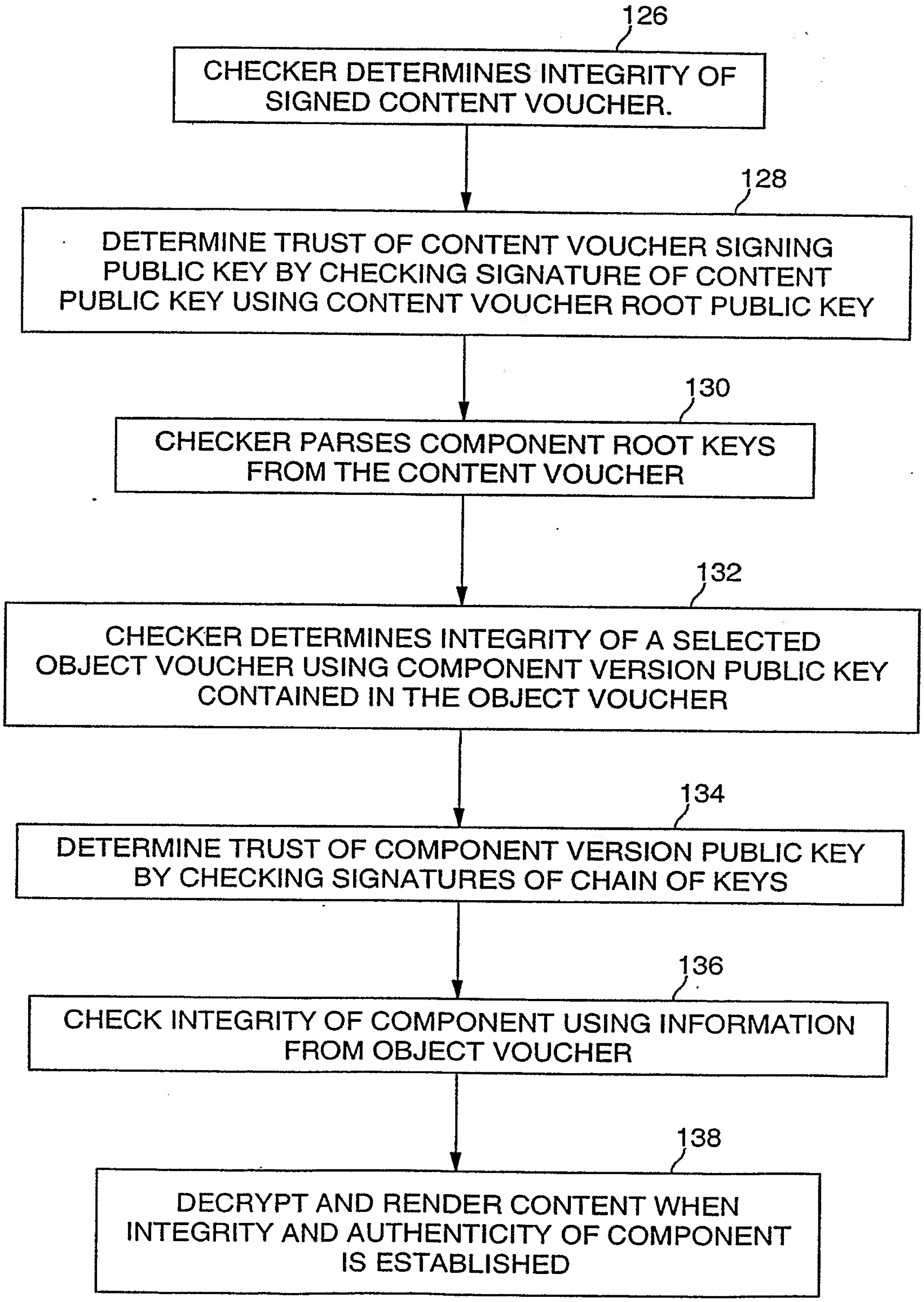

126
CHECKER DETERMINES INTEGRITY OF SIGNED CONTENT VOUCHER.
128
DETERMINE TRUST OF CONTENT VOUCHER SIGNING PUBLIC KEY BY CHECKING SIGNATURE OF CONTENT PUBLIC KEY USING CONTENT VOUCHER ROOT PUBLIC KEY
130
CHECKER PARSES COMPONENT ROOT KEYS FROM THE CONTENT VOUCHER
132
CHECKER DETERMINES INTEGRITY OF A SELECTED OBJECT VOUCHER USING COMPONENT VERSION PUBLIC KEY CONTAINED IN THE OBJECT VOUCHER
134
DETERMINE TRUST OF COMPONENT VERSION PUBLIC KEY BY CHECKING SIGNATURES OF CHAIN OF KEYS
136
CHECK INTEGRITY OF COMPONENT USING INFORMATION FROM OBJECT VOUCHER
138
DECRYPT AND RENDER CONTENT WHEN INTEGRITY AND AUTHENTICITY OF COMPONENT IS ESTABLISHED

Figure 6

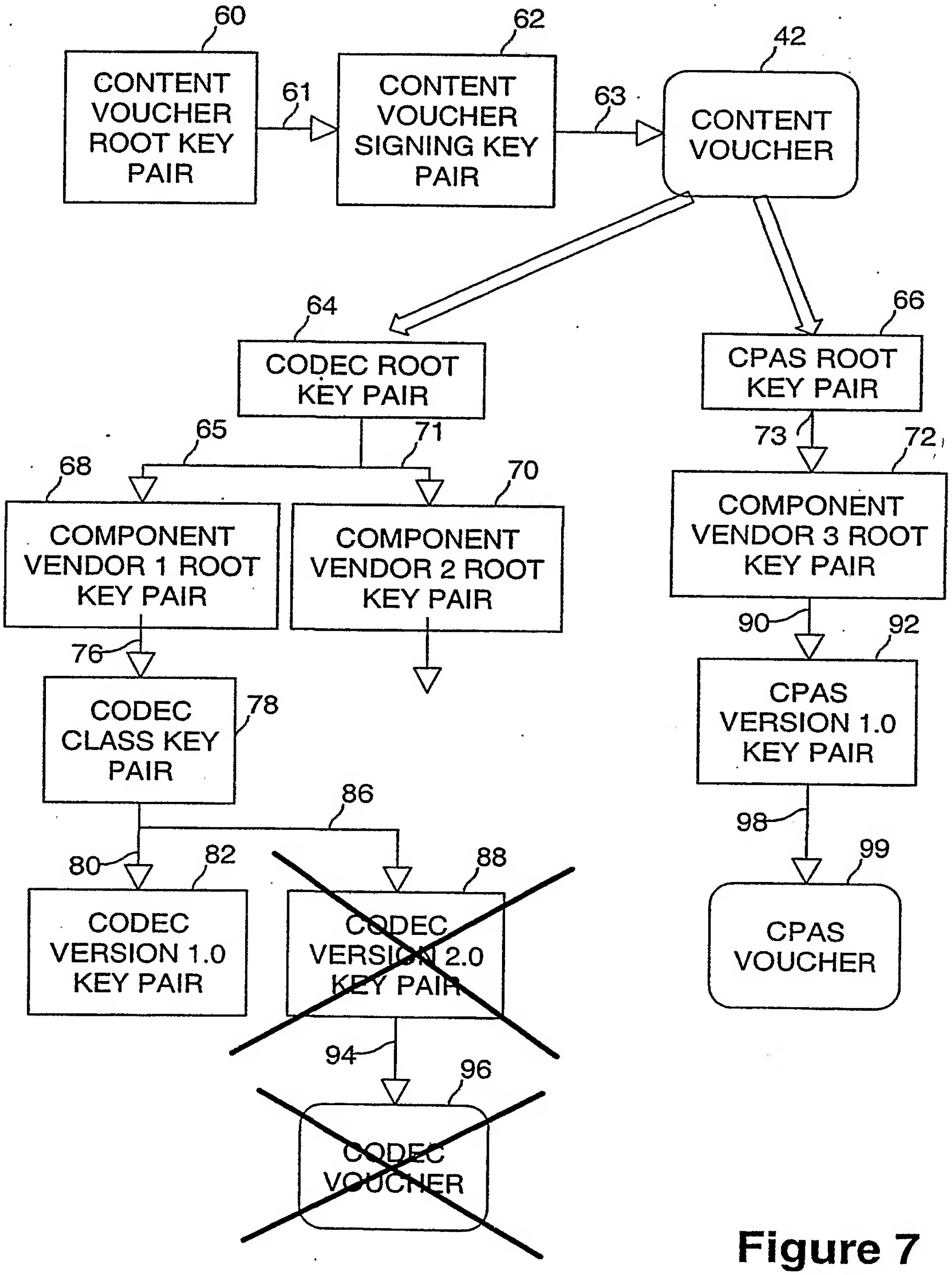

60
CONTENT VOUCHER ROOT KEY PAIR
61
62
CONTENT VOUCHER SIGNING KEY PAIR
63
42
CONTENT VOUCHER
64
CODEC ROOT KEY PAIR
66
CPAS ROOT KEY PAIR
65
71
73
72
68
COMPONENT VENDOR 1 ROOT KEY PAIR
70
COMPONENT VENDOR 2 ROOT KEY PAIR
COMPONENT VENDOR 3 ROOT KEY PAIR
76
90
92
CODEC CLASS KEY PAIR
78
CPAS VERSION 1.0 KEY PAIR
86
98
80
82
88
99
CODEC VERSION 1.0 KEY PAIR
CODEC VERSION 2.0 KEY PAIR
CPAS VOUCHER
94
96
CODEC VOUCHER

Figure 7

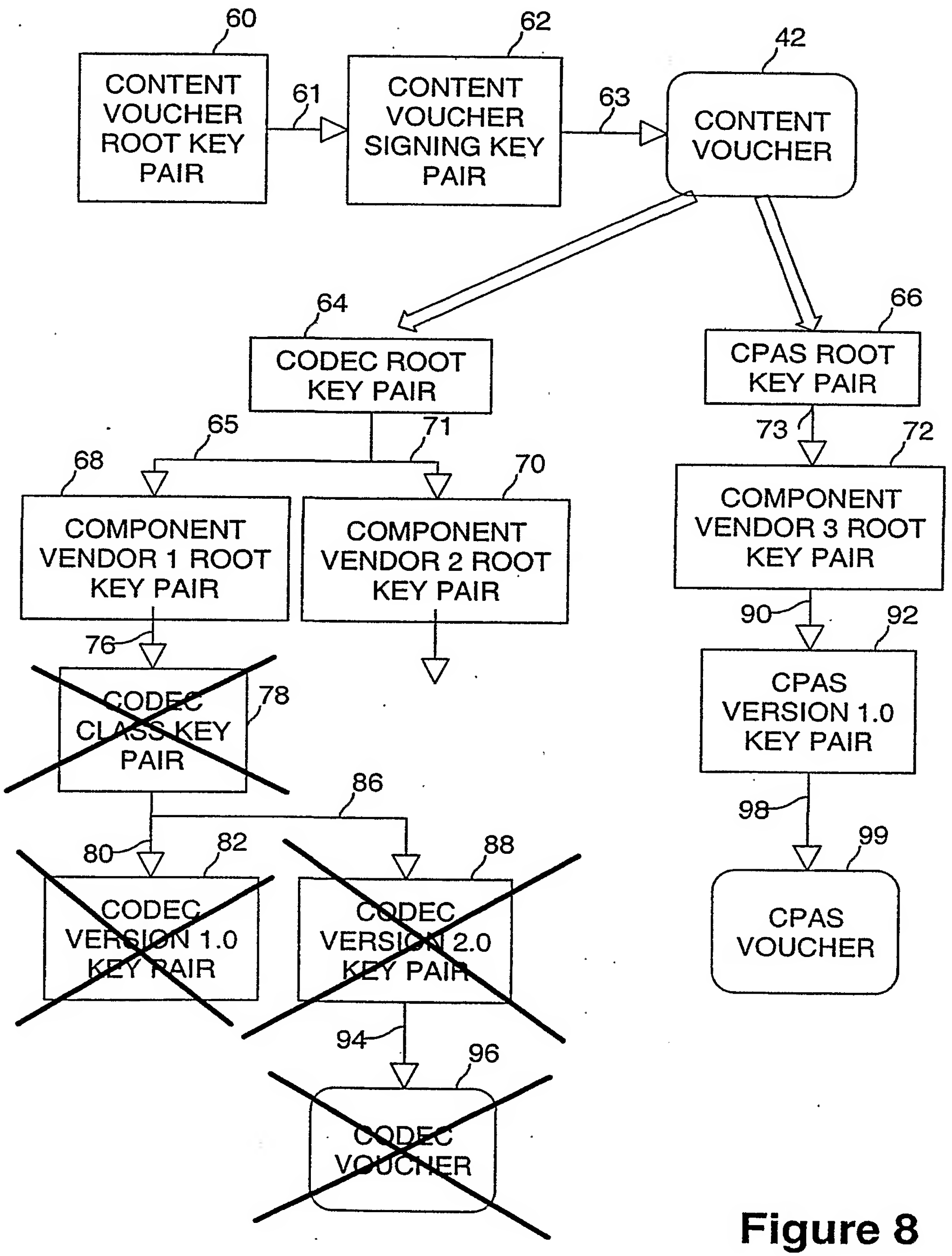

60
CONTENT VOUCHER ROOT KEY PAIR
61
62
CONTENT VOUCHER SIGNING KEY PAIR
63
42
CONTENT VOUCHER
64
CODEC ROOT KEY PAIR
66
CPAS ROOT KEY PAIR
65
71
68
70
73
72
COMPONENT VENDOR 1 ROOT KEY PAIR
COMPONENT VENDOR 2 ROOT KEY PAIR
COMPONENT VENDOR 3 ROOT KEY PAIR
76
90
92
CODEC CLASS KEY PAIR
78
CPAS VERSION 1.0 KEY PAIR
86
98
80
82
88
99
CODEC VERSION 1.0 KEY PAIR
CODEC VERSION 2.0 KEY PAIR
CPAS VOUCHER
94
96
CODEC VOUCHER

Figure 8

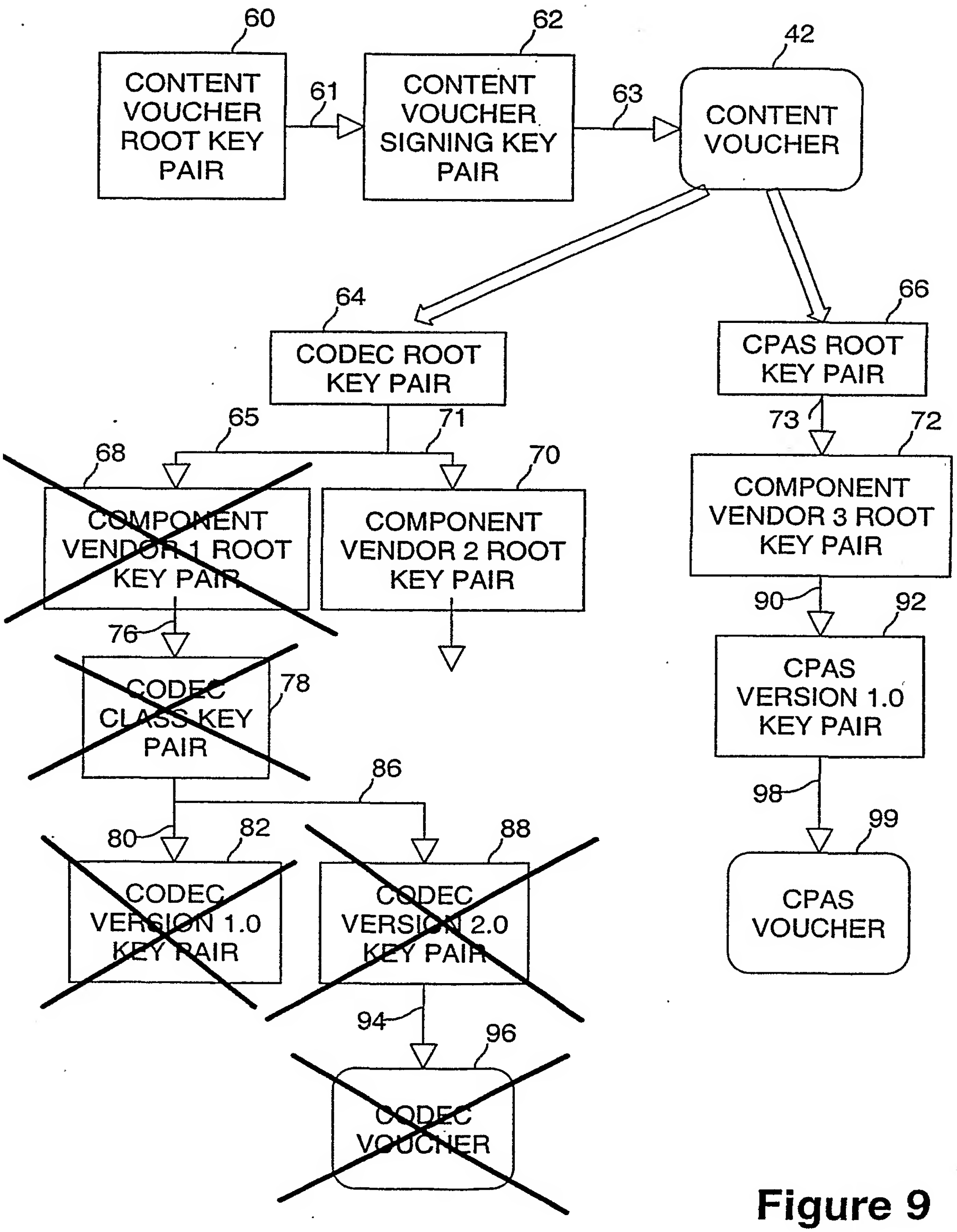

60
CONTENT VOUCHER ROOT KEY PAIR
61
62
CONTENT VOUCHER SIGNING KEY PAIR
63
42
CONTENT VOUCHER
64
CODEC ROOT KEY PAIR
66
CPAS ROOT KEY PAIR
65
71
68
COMPONENT VENDOR 1 ROOT KEY PAIR
70
COMPONENT VENDOR 2 ROOT KEY PAIR
73
72
COMPONENT VENDOR 3 ROOT KEY PAIR
76
CODEC CLASS KEY PAIR
78
90
92
CPAS VERSION 1.0 KEY PAIR
86
80
82
CODEC VERSION 1.0 KEY PAIR
88
CODEC VERSION 2.0 KEY PAIR
98
99
CPAS VOUCHER
94
96
CODEC VOUCHER

Figure 9

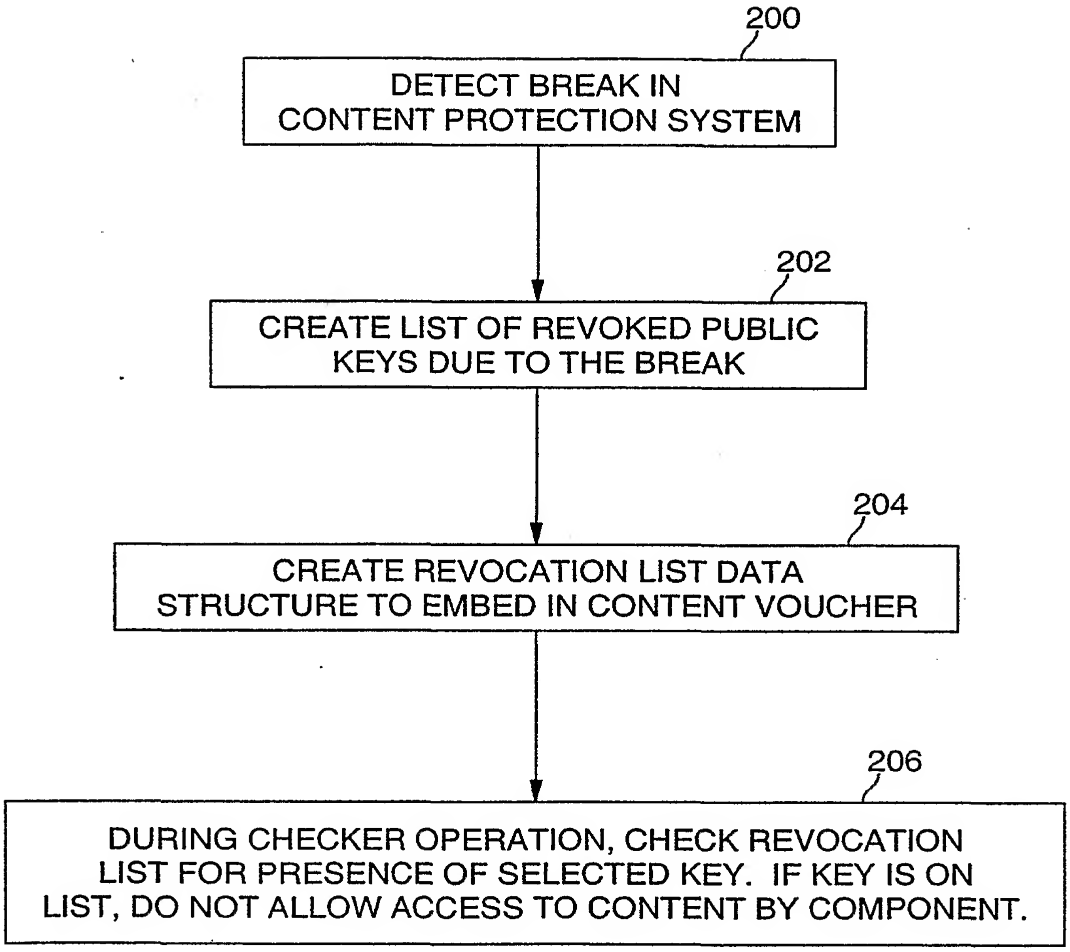

200
DETECT BREAK IN CONTENT PROTECTION SYSTEM
202
CREATE LIST OF REVOKED PUBLIC KEYS DUE TO THE BREAK
204
CREATE REVOCATION LIST DATA STRUCTURE TO EMBED IN CONTENT VOUCHER
206
DURING CHECKER OPERATION, CHECK REVOCATION LIST FOR PRESENCE OF SELECTED KEY. IF KEY IS ON LIST, DO NOT ALLOW ACCESS TO CONTENT BY COMPONENT.

Figure 10

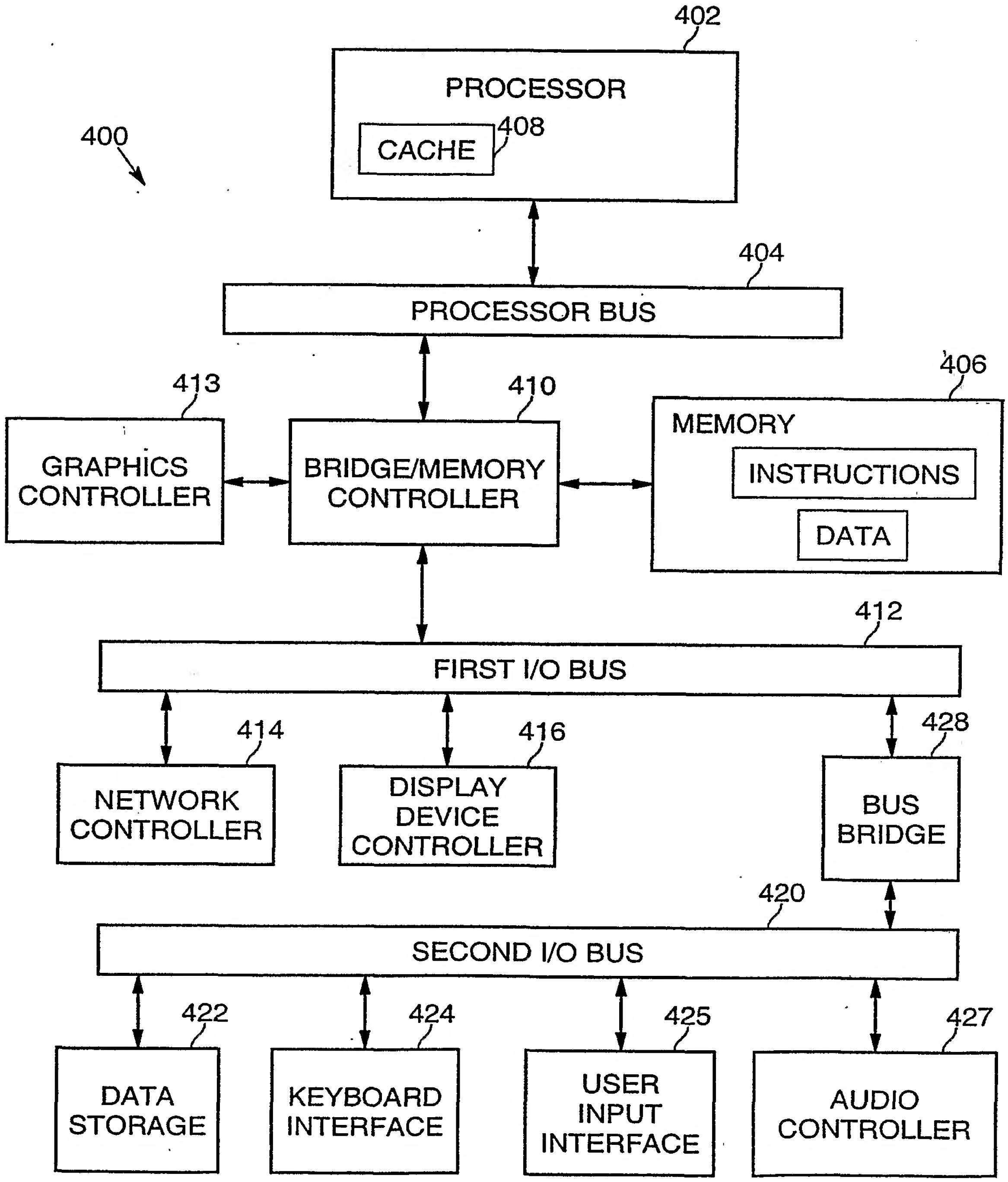

402
PROCESSOR
CACHE
408
400
404
PROCESSOR BUS
413
410
406
GRAPHICS CONTROLLER
BRIDGE/MEMORY CONTROLLER
MEMORY
INSTRUCTIONS
DATA
412
FIRST I/O BUS
414
416
428
NETWORK CONTROLLER
DISPLAY DEVICE CONTROLLER
BUS BRIDGE
420
SECOND I/O BUS
422
424
425
427
DATA STORAGE
KEYBOARD INTERFACE
USER INPUT INTERFACE
AUDIO CONTROLLER

Figure 11